# Strsafe.h

Daniel Plakosh, Software Engineering Institute [vita[1]]



2005-09-27; Updated 2008-10-06 L2 / D/P, L[2]

Microsoft provides a set of safer string handling functions for the C programming language called Strsafe.h. These functions are intended to replace their built-in C/C++ counterparts, as well as any legacy Microsoft-specific string handing functions.

## Development Context

String manipulation

## Technology Context

C/C++, Win32

## Attacks

Attacker executes arbitrary code on machine with permissions of compromised process or changes the behavior of the program.

## Risk

Standard C string manipulation functions are prone to programmer mistakes that can result in buffer overflow vulnerabilities.

## Description

Microsoft provides a set of safer string handling functions for the C programming language called Strsafe.h [MSDN 08]. These functions are intended to replace their built-in C/C++ counterparts, as well as any legacy Microsoft-specific string handing functions.

The Strsafe functions support both ANSI and Unicode characters, always return a status code, and require that the programmer always specifies the size of the destination buffer. Separate functions are provided that allow the programmer to specify the size of the destination buffer using either character or byte counts.

The Microsoft Strsafe library functions guarantee that all strings are null terminated (even if they are truncated) and that a write does not occur past the end of the destination buffer. These functions are safe as long as the programmer inputs the actual starting address of the destination buffer and correct length. As a result, care must still be taken when using these functions.

Figure 1 shows an example program that performs a secure string copy on line 6 and a secure string concatenation on line 11.

**Figure 1. Microsoft Strsafe example**

```
1. #include <Strsafe.h>
2. int main(int argc, char *argv[])
3. {
4.   char MyString[128];
5.   HRESULT Res;
6.   Res=StringCbCopy(MyString, sizeof(MyString), "Program 1. Name is ");
7.   if (Res != S_OK) {
```

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/268-BSI.html (Plakosh, Daniel)

```
 8.     printf("StringCbCopy Failed: %s\n", MyString)
 9.     exit(-1);
10.   }
11.   Res=StringCbCat(MyString,sizeof(MyString),argv[0]);
12.   if (Res != S_OK) {
13.     printf("StringCbCat Failed: %s\n", MyString);
14.     exit(-1);
15.   }
16.   printf("%s\n", MyString);
17.   return 0;
18. }
```

It is also important to remember that the Strsafe functions, such as `StringCchCopy()` and `StringCchCat()`, do not have the same semantics as the Microsoft CRT functions `strncpy_s()` and `strncat_s()`. When `strncat_s()` detects an error it sets the destination string to a null string, while `StringCchCat()` fills the destination with as much data as possible and then null-terminates the string.

### References


| | |
|---|---|
| [ISO/IEC 99] | ISO/IEC. *ISO/IEC 9899 Second edition 1999-12-01 Programming languages — C*. International Organization for Standardization, 1999. |
| [MSDN 08] | Microsoft Corp. *Using the Strsafe.h Functions*[19] (2008). |


## Pearson Education, Inc. Copyright